Word Watch: An Application for Creating Read-Word Data for Subvocal Recognition Tasks

George Toumbas

Advisor: Dr. Felix Heide

Senior Thesis

Department of Computer Science
Princeton University
Spring 2023

Abstract

Subvocal Recognition (SVR) has the potential to revolutionize communication for those with speech impairments and enhance human-technology interactions. However, current data collection methods for SVR are time-consuming, laborious, and difficult to scale. This paper presents Word Watch, an application that offers a novel approach to SVR data collection by targeting subvocalizations during reading. Word Watch employs a commercially available eye-tracker, calibrated text-settings, and custom algorithms to predict which words a user has read and when those words were subvocalized. Evaluation of the Word Watch system shows that it can accurately identify which words have been read at a reasonable reading pace. Further research, however, is needed to validate the application's estimates of when read-words are subvocalized.

1. Introduction

Subvocal Recognition (SVR), the ability to understand a person's internal, silent speech, is a problem that has been studied for decades. If mastered, SVR has the potential to radically improve the lives of individuals with speech impairments and augment our everyday interactions with technology. Although researchers have made impressive progress in recent years, most SVR systems are still far from being practical. A major obstacle to the development of practical SVR systems is a lack of robust data collection methods. Often, the process of collecting data looks something like this: A researcher prompts a participant to think a certain word or sentence, records brain, muscle, or other signals, and then stops recording. This process creates data points of biological signals and corresponding text. A researcher can then use this data to train a machine learning model to predict text from signals. This method of dataset creation is extremely time-consuming, tedious for participants, and challenging to conduct at scale. In an attempt to address these limitations, this project introduces Word Watch, an application for collecting data for SVR. Word Watch presents a novel approach to SVR data collection that targets subvocalizations that occur during reading. To accomplish this, Word Watch utilizes a commercially available eye-tracker, analyzes data from this tracker while a user is reading, and creates time-series data comprised

of read-words and estimates of when those words were subvocalized. The hope is that this data can then be used in conjunction with biological data in order to train machine learning models attempting to perform SVR. To increase the accuracy of the read-word data, Word Watch features custom calibrations, algorithms, and an in-built reading interface. Although further research is needed to evaluate accuracy of our subvocalization timing estimates, we hope that Word Watch can aid researchers in exploring alternative, faster methods of data collection for SVR.

2. Problem Background

2.1. Tracking Reading Progress Using Eye Tracking

There have been many studies which utilize eye-tracking technology to study reading. However, there have not been, to the best of our knowledge, studies that attempt to identify reading progress in the way that Word Watch aims to. There have been some attempts to track reading progression using eye-trackers. One such example is the paper *An Approach to Track Reading Progression Using Eye-Gaze Fixation Points* by Bottos et al. The goal of this paper was to use commercially available eye-trackers to accurately track the line being read by an individual without any prior knowledge of the text layout. The study used the Gazepoint GP3 eye-tracker to collect eye-gaze data from a single test subject, a male in his twenties. The researchers developed a data collection program in Python to communicate with the eye-tracker. During data collection, only a single line of text was displayed on the screen at a time, allowing the researchers to accurately record the ground truth. The test subject was asked to read 25 lines of text, with the eye-tracker logging eye-gaze fixation coordinates at 60 Hz. To address the noisy nature of the eye-tracking data, the researchers employed hidden Markov models (HMMs) for line detection. Their system achieved an average error of about 16.9% [1]. Although Word Watch is also attempted to track reading progress, its goals and constraints are importantly different. For one, we are attempting to use eye-tracking data to predict when specific words, not lines, are read. We are also not constrained by a lack of knowledge about the spacing and formatting of text. Word Watch is aware of and modifies text settings.

2.2. Subvocal Recognition

2.2.1. Early Research and NASA's EMG-based SVR

As machine learning models have become more powerful, there has been an increase in the amount of research done in SVR. Many modern SVR techniques leverage electromyography (EMG), a technology used to measure the electrical activity of muscles. SVR research has shown that the electrical activity of facial, throat, and laryngeal muscles can be used to predict which word a person is currently subvocalizing. Intuitively, this makes some sense, as all of these muscles are engaged during normal, vocalized speech. One of the first efforts which demonstrated the viability of using EMG data for SVR was led by a group of NASA researchers. Their 2005 paper, titled *Web Browser Control Using EMG Based Sub Vocal Speech Recognition*, showed that SVR of complete words could be accomplished. The researchers utilized 5 EMG electrodes, placed "on the side of the throat near the larynx and under the chin." They then recorded signals while participants thought of either one of five control words ("Stop, Go, Left, Right, Alpha") or a number (0-9). They recorded 200 samples of each word from five subjects. Using a scaled conjugate gradient neural network, the researchers were able to correctly classify the five control words with an accuracy of 92% and each digit with an accuracy of 72% [2].

2.2.2. MIT Media Lab's AlterEgo Project

A more recent and widely publicized attempt at SVR is the MIT Media Lab project called AlterEgo. The project's first paper, *AlterEgo: A Personalized Wearable Silent Speech Interface*, was published in 2018. The project involved the development of a custom wearable headset, removing the need for a bulky, medical-grade EMG device. The AlterEgo wearable device used in this study has seven electrodes strategically placed on different muscles in the face. The AlterEgo team first conducted a pilot study where they asked three participants to silently subvocalize the words "yes" and "no." The data collected from this pilot study consisted of five hours of internally vocalized text. This data led the researchers to select the laryngeal region, the hyoid region, levator anguli oris, orbicularis oris platysma, and the anterior belly of the digastric mentum as rough electrode locations.

In their main study, the researchers chose to use six limited vocabulary sets. Each set consisted

of no more than 15 unique words. Across these vocabulary sets, the AlterEgo team recorded approximately 31 hours of silently spoken text recorded from the same three participants as the pilot study. The data was preprocessed with various transformations before being input into a simple neural network architecture consisting of a 1D signal, three convolutional ReLU layers, a fully connected layer, and a final classification layer. The largest vocabulary set consisted of the numbers 0-9 and the words "multiply", "add", "subtract", "divide", and "percent". On this set of words, their classifier correctly classified subvocalizations with a median accuracy of 92% across ten users [3].

The AlterEgo team has continued to work on achieving more reliable SVR. The group wrote a second paper in 2020, *Non-Invasive Silent Speech Recognition in Multiple Sclerosis with Dysphonia*. As the title suggests, this paper was focused on performing SVR for individuals with MS. The study involved three patients and used a more traditional EMG machine for data collection, with eight electrode placements instead of seven. Initially, the electrode positions were the same as in the previous paper, but were slightly tweaked as the study progressed. The electrodes were sampled at 250HZ. Importantly, unlike the 2018 paper, this study attempted to classify subvocalized sentences rather than the subvocalization of individual words. Training data was collected from three patients suffering from MS and consisted of 15 distinct sentences. All patients subvocalized each sentence ten times. To improve sentence cutoffs, the study included a GUI of EMG data that allowed patients to press a button in order to indicate when they had stopped subvocalizing a sentence. Additionally, heartbeat artifacts and high frequency noise were removed from the data before training. Unlike the 2018 study, three models were trained, each one using the data from an individual patient. Each model was a 16-layer convolutional neural network. The three models had accuracies of 79%, 87%, and 77% [4].

2.2.3. Generating Audio Signals from EMG Data

There has also been SVR research that does not involve classifying EMG data into text categories. In their 2020 paper titled *Digital Voicing of Silent Speech*, Gaddy and Klein attempted to generate understandable audio signals directly from EMG data. Importantly, whereas previously discussed research aimed to predict text from subvocalized thoughts, this paper attempts to generate audio

from EMG data collected while a participant is silently mouthing speech. The paper utilizes vocalized speech, EMG from vocalized speech, and EMG from silently mouthed speech collected from 8 electrodes on the face and neck. The researchers recorded 20 hours of facial EMG signals from a single speaker. Unlike the AlterEgo and NASA studies, this study included both a closed and open vocabulary settings. In the closed setting, there were a total of 67 words, 25 minutes of silent speech, 30 minutes of vocalized speech, with 500 utterances and 4 words per utterance. However, most data was recorded with an open vocabulary, collected as the participant was reading books from Project Gutenberg. After generating the audio, it was then evaluated by having another participant transcribe the speech and measuring the Word Error Rate of that transcription (lower is better). Gaddy and Klein achieved a WER of 3.6 in the closed vocabulary setting and a WER of 68 in the open vocabulary setting [5]. In a 2021 follow-up paper, *An Improved Model for Voicing Silent Speech*, Gaddy and Klein were able to improve open vocabulary performance on the same data-set to a WER of 42 through architectural improvements [6].

2.3. Subvocalizations During Reading

2.3.1. Subvocalization and Reading Comprehension

Word Watch assumes that subvocalization occurs during reading, or if it does not occur naturally for a user, he or she has the ability to subvocalize during reading. There has been considerable research about subvocalizations which occur during reading. One of the most cited works in this area of research is the 1980 study *Subvocalization and Reading for Meaning*. The study, conducted by researchers at the University of Massachusetts Amherst, investigates how the suppression of silent speech during reading impacted reading comprehension. To test this, the researchers compared the reading and listening comprehension ability of participants who were simultaneously counting or repeating a phrase out loud. The comprehension ability of these two groups of participants was then compared to that of two other groups who either read silently or listened without distraction. Sixty-four undergraduate students participated in this study, with each of the four groups containing sixteen students. Each group was presented with either 8 different aural or written stories, and then

asked a yes or no question about each one. The group silently reading performed roughly 10% better than the distracted group. While there could be confounding variables in this approach, it suggests that subvocalizations play some role in reading comprehension [7].

2.3.2. Eye Movements and Subvocalizations

Understanding the relationship between eye-movements that occur during reading and subvocalizations is important for estimating the timings of specific subvocalizations. Unfortunately, there has not been, to the best of our knowledge, research which explicitly studies this relationship. There has, however, been research which examines the relative timing of fixations and vocalized speech during reading. There is also some reason to think that this relationship bears a resemblance to the one between fixations and subvocalizations. Fixations are brief periods of time when the eyes remain relatively still and focused at a particular location. During reading, our eyes jump from fixation point to fixation point. These jumps are known as saccades [8]. The 2015 paper *The eye-voice span during reading aloud* investigates the timing differences between fixations on words and the subsequent articulation of those words. The study monitored the eye movements of thirty-two subjects (12 males, 20 females) while they each read 144 sentences out loud. The eye movements of another 31 subjects (12 males, 19 females) were monitored while they read the same sentences silently. Eye movements were monitored using an Eyelink 1000. The study found that the mean eye-voice span, or the time between the first fixation on a word to the beginning of its articulation, was 561 milliseconds with a standard deviation of 230ms. In the group that read aloud, the average mean fixation duration was 253 ms with a standard deviation of 96 ms, while the group that silently read had a mean fixation duration of 209 ms with a standard deviation of 81 ms. The researchers note that, although there were some differences between the oral and silent groups, the eye movements of the two modes are similar in many ways. Furthermore, drawing on these results and findings from other research, they tentatively state that subvocalization during silent reading may occur during the fixation on the subsequent word [9].

3. Approach

3.1. Hardware

For our eye-tracking hardware, we chose to use the Pupil Core eye tracker made by Pupil Labs. Eye-trackers generally fall into two categories: screen-based and head-mounted eye trackers. Screen-based eye trackers use cameras mounted on or around a computer screen to track the movement of a user's eyes and estimate where on the screen a user is looking. Head-mounted eye trackers usually take the form of glasses and use cameras pointed at the eyes in conjunction with one or more outwards facing cameras to estimate where in a scene a user is looking. The Pupil Core is a head-mounted eye tracker. It consists of two eye-facing infrared cameras and one scene-facing camera. The two eye cameras record at 200Hz and at a resolution of 192x192px and the scene camera has a resolution of 30Hz at 1080P, 60Hz at 720p, and 120Hz at 480p. The Pupil Core has an accuracy of 0.60° and a precision of 0.02° [10].

The Pupil Core offers several advantages over other similarly capable eye-tracking solutions. Costing roughly $3000, it is cheaper than other screen-based eye-trackers typically used in research. Tobii, perhaps the most prominent company in the eye-tracking space, has research-grade eye-trackers ranging from $18,900 dollars to $4,200 dollars. Their cheapest research device, the Tobii Pro Nano, has a worse sampling frequency and precision than the Pupil Core at 60Hz and a 0.10° RMS. It has a slightly better accuracy, however, of 0.3° [11].

The Pupil Core comes with easy to use software, Pupil Capture and Pupil Player, that work on all platforms (Windows, Mac, and Linux). The Pupil Capture software that comes with the Pupil Core is easy to use and provides a visual representation of the eye-tracking data being collected in real-time. Importantly, Pupil Capture allows fast calibration using a 5 point calibration system [12]. The most notable advantage of the Pupil Core is the fact that all of its software is completely open-source. Most other eye-trackers, including those made by Tobii, use proprietary software. The Pupil Core has extensive documentation that makes it quite easy to build applications that utilize data coming from the eye-tracker. The Pupil Core's open-source status has allowed an ecosystem of

plugins to develop for the device's software, Pupil Capture.

3.2. Software Used

In addition to using Pupil Capture for calibration, we also make use of its Surface Tracking [13] and Network API [14] plugins. Both of these plugins were developed by Pupil Labs. Before using Word Watch, users are required to calibrate the Pupil Core within Pupil Capture, register and name a surface using the Surface Tracking plugin, and enable data broadcasting through the Network API on their desired IP address and Port. Word Watch requires Pupil Capture to be running in order to function. The Surface Plugin and Network API Plugin only need to be configured once, while the Pupil Capture calibration needs to be done before every session of Word Watch.

We chose to build Word Watch using the Python programming language. We did this for several reasons. For one, Python is extremely easy to use and consequently lends itself to rapid development. Python is also cross-platform, allowing Word Watch to be used by researchers who posses a variety of devices. To interact with the user, we chose to use Tkinter, Python's standard interface to the TK Graphical User Interface (GUI) toolkit [15]. There are also several 3rd-party Python packages that are crucial for Word Watch:

- **CustomTkinter:** This package updates elements of Tkinter to allow for the creation of more modern-looking user interfaces. All the buttons, pages (known as frames in Tkinter), and fields for user input utilize CustomTkinter [16].
- **Tkinter-tooltip:** This package allows Word Watch to display tooltip messages when a user hovers over buttons on the navigation bar [17].
- **Pupil_apriltags:** This package is made by Pupil Labs. We use it in Word Watch to detect April Tags, allowing us to determine the on-screen location of the surface detected by the Surface Tracking plugin within Pupil Capture [18].
- **ZMQ** and **Mspack**: The ZMQ package is necessary for communicating with Pupil Capture's Network API plugin. The Mspack package is used for manipulating the data received from the Network API plugin [19][20].

- **Numpy** and **Scikit-learn**: Numpy is used for the storage and manipulation of gaze and fixation data, while Scikit-learn is used in some of the calculations made during in-app calibrations [21][22]
- **Colour:** This package is used for ease of color manipulation in the Word Watch interface [23].
- **Fitz** pdf package: This package [24] was used to extract PNG images of April Tags from PDFs made by AprilRobotics [25].

3.3. Surface Tracking

Before we can attempt to track reading, we must first determine where on the screen a user is looking. To accomplish this, we need to determine the on-screen coordinates of a user's gaze based on the scene-facing camera of the Pupil Core. We utilize the Surface Tracking plugin to do this. The plugin allows 2d surfaces present in the view of the Pupil Core's world camera to be tracked. In order to track a surface, a user must register the surface presented in Word Watch in the Pupil Capture application. This involves selecting April Tags in the view of the scene camera and creating a rectangle around them to define the surface. After naming the surface, the plugin calculates the normalized coordinates of gazes and fixations on the surface. We then convert these surface coordinates to on-screen coordinates and in-app coordinates, which we then use to monitor a user's reading.

3.4. In-App Calibrations

The Pupil Core, like all eye-trackers, lacks perfect accuracy, and the conversion to surface coordinates can further decrease accuracy. Word Watch includes in-app calibrations that aim to compensate for these inaccuracies. Here, we will broadly discuss the purpose of the calibrations. Each calibration is described in more detail in implementation section. The calibration system is composed of three stages. The first calibration presents users with points to look at, similar to the Pupil Capture calibration, and then calculates the accuracy of on-screen gaze and fixation coordinates in pixels. The word and line spacing to be used when reading are then adjusted based on these values. The second calibration is similar to the first, but it instead presents individual words

of varying sizes for the user to read. It then predicts which words were read and adjusts text spacing further if the prediction accuracy is too low. We perform these gradual increases in text spacing in an attempt to find the minimum text spacing that allows for accurate read-word predictions. While we could have decided to simply initially set the text spacing to be conservatively large, we felt that this would make the experience of reading more uncomfortable.

While the first two calibrations are aimed at fine-tuning text settings, the third calibration attempts to compensate for differences in saccade sizes. When reading, fixation points do not perfectly correspond to individual words. Around one third of the time, a person will read multiple words during a single fixation. This means that some words, although they have been read, are never directly fixated on [26]. To account for this, Word Watch has an algorithm (see 3.6.2) that uses a saccade size threshold to determine when gaps between fixations represent reading and when they simply represent a user skipping over text. The third calibration measures a user's average horizontal saccade size and then selects a threshold to be used in future read-word predictions. To do this, the third calibration highlights multiple random sentences for the user to read. The size of the gaps between fixation points on the highlighted words are then measured in pixels and averaged. Finally, five round of predictions are made with varying saccade thresholds slightly below, slightly above, and at the measured average saccade size. The threshold that results in the best read-word classifications is then stored to be used in data collection.

3.5. Data Output

The read-word data is stored in a JSON file where the application was run. Times are in Unix Time [27]. The file includes the following information for each word that the user has read:

- **Word**: The text of the word that was read
- **Start Time**: Estimate of when the user started subvocalizing the word
- **End Time**: Estimate of when the user finished subvocalizing the word
- **X**: The x-coordinate (px) of the fixation on the screen.

- **Y**: The y-coordinate (px) of the fixation on the screen.
- **Unique Fixation**: A boolean value that is true when the word was the only one read during its corresponding fixation and false otherwise.

3.6. Read-Word Calculation

In this section, we will describe how Word Watch identifies which words a user has read. We have a two-step approach for generating our read word-predictions. The first step uses only fixation data to identify read-word candidates. The second step uses these candidates and a saccade size threshold set in calibration to predict which fixation points correspond to multiple read words. We do not assume that a user will read every sentence on every page. We do this to allow users to only read desired sections of the text. To simplify our calculations, we do assume, however, that words will be read in order without rereading and that users will tend to read complete sentences.

3.6.1. Predictions Using Fixation Data

The first step of our approach relies only on fixation data received from the Pupil Core. With the Pupil Core, a fixation data point is created when a user looks at a specific point for more than 100ms [28]. After a reading session has been completed, we examine all recorded fixation points and identify the closest word to each point, tentatively marking these words as read. We remove any read words that are part of a sentence where fewer than 20% of the words are marked read. Although this outlier removal does result in false negatives, we decided that avoiding false positives was more important for our goal of creating usable data. As seen in Figure 1, using fixation data by itself can often capture a significant number of read-words. It often, however, fails to capture all read-words.

Figure 1: A simple example of read-word candidates created from fixation data alone. In this case, the entire sentence has been read. The green words have been fixated on and have been selected as read-word candidates.

3.6.2. Saccade Accommodation

Algorithm 1 Multi-Word-Single-Fixation (MWSF) Algorithm

```
Input: Text, Read-word candidates, Saccade threshold
for every word in the text (in order they appear) do
    if word is non-read and directly follows read-word then
        follow contiguous non-read words until read-word
        if the sum of characters in contiguous non-read words ≤ Saccade threshold then
            mark these contiguous non-read words as read
        end if
        if non-read streak at end of sentence without closing read word then
            if streak ≤ threshold AND more than 60% of words in current sentence are marked read
            then
                mark these non-read words as read
            end if
        end if
    end if
end for
for every sentence in text do
    if sentence starts with non-read word then
        follow contiguous non-read words until read-word
        if the sum of characters in contiguous non-read words ≤ Saccade threshold AND more
        than 60% of words in current sentence are marked read then
            mark these contiguous non-read words as read
        end if
    end if
end for
```

The Multi-Word-Single-Fixation (MWSF) Algorithm 1 is the main component of this step in read-word identification. A visualization of its effects be seen in Figure 2. After the MWSF Algorithm is executed, two additional validations are performed in an effort to remove remaining false positives. First, we once again check for sentences that have low-rates of read-words. All read words that belong to a sentence where fewer than 20% of words are read are removed. Finally, we confirm the order in which sentences were read. For each sentence with read-words, we calculate the average start time of fixations on those read-words. We then confirm that this average start-time is later than those of all previous sentences with read words. If it is not (i.e this sentence was read before a previous sentence), we remove all read-words from that sentence. In practice, this final step removes false positives which sometimes occur when fixation points are too low, resulting in read words on the line below the one currently being read.

Figure 2: A visualization of the steps taken during the Multi-Word-Single-Fixation (MWSF) Algorithm on a simple sentence. In this case, the saccade threshold is at least seven characters. Words marked read using fixation data alone are in green and words identified as read by the algorithm are marked in blue.

3.6.3. Subvocalization Time Estimation

All time estimates of subvocalizations are created after final read-word predictions have been made. There is surprisingly little research that investigates the delay between fixation start times and subvocalization start times. The heuristics illustrated in Figure 3, while loosely based on the research which does exist [9], are likely overly simplistic. We were unable, for instance, to find research which discusses the relationship between fixation times and subvocalization times when a

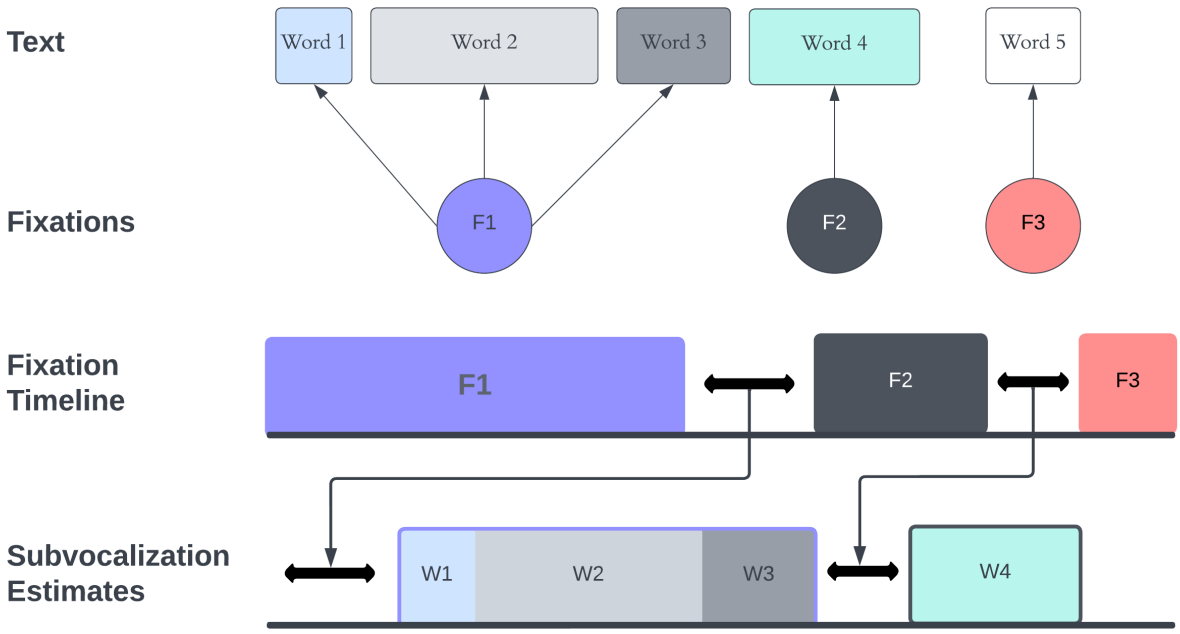

Figure 3: This Figure illustrates how subvocalization times are estimated in a sentence where every word is marked as read. It shows cases where a single fixation corresponds to multiple read words (F1) and where a single fixation corresponds to a single read word (F2, F3). The duration of each word's subvocalization is estimated by taking into account the character length of the word as a proportion of the corresponding fixation duration. Finally the starts of subvocalizations are calculated by using the time difference between subsequent fixations of adjacent read words.

fixation corresponds to multiple subvocalizations. Currently, we assume that duration of a fixation is directly linked to the duration of a subvocalization. This can be seen in the Subvocalization Estimates component of Figure 3. Here, the sum of the durations of the subvocalizations of Words 1, 2, and 3 equal the duration of Fixation 1, with the length of each individual subvocalization being proportional to the length of its corresponding word. To account for the time it takes for the brain to process information before a subvocalization, we offset start time of a subvocalization (W4), or a set of subvocalizations (W1, W2, W3) by the delay before the next fixation in a sequence of contiguous words marked as read. In the case where there is a gap between read words, we offset the start of a subvocalization from its fixation by the average delay between fixations in contiguous read words across an entire session. For example, if F3 did not exist and Word 5 was marked as unread, the offset between subvocalization W3 and W4 would be equal to the delay between F1 and F2. In this simple case, the average delay between fixations of contiguous read words is simply the

delay between F1 and F2. The problems with the assumptions of this approach are discussed in the Limitations section.

4. Implementation

4.1. Application Overview

4.1.1. Navigation Bar:

As seen in Figure 4, the navigation bar is on the left side of the Word Watch window. The navigation bar is the only component of Word Watch that remains visible at all times. It consists of four buttons and two status indicators. All button icons were sourced from a free website [29]. Additionally, all buttons display the button names in a tooltip when the mouse hovers over them. From top to bottom, the buttons are as follows:

- **The Home Button**: This button displays the Home page, where a user can configure their connection to the Pupil Core.
- **The Calibrations Button**: This button takes the user to the Calibrations page, where a user can sequentially take the three calibrations.
- **The Test Read-Word Predictions Button**: This button can only be clicked once all calibrations have been completed. It takes the user to a page where they can evaluate the accuracy of the read word predictions and determine if they want to retake any calibrations.
- **The Monitor Reading Button**: This button can only be clicked once all calibrations have been completed. This button takes the user to the reading monitoring page, where users can actually collect read-word data.
- **The Text Settings Button**: This button displays a Text Settings window that allows users to manually change how text is displayed.

The two status indicators are located at the bottom of the navigation bar. The top-most status icon informs the user if their configuration of the Pupil Capture Network API plugin is correct and Word Watch is successfully receiving gaze and fixation data from the Pupil Core. The bottom status icon informs the user if they have correctly registered Word Watch's April Tags in Pupil Capture's

Surface Tracking plugin. Each status indicator has four possible states: *Not Connected*, *Attempting to Connect*, *Failed To Connect*, and *Successfully Connected*. When hovered over, each status icon will display the current state in a tooltip. The colors of the status icons also change depending on state. The possible corresponding colors for the listed states are gray, orange, red, and green, respectively. The status indicators are updated via two buttons found on the Home Page.

4.1.2. Home Page:

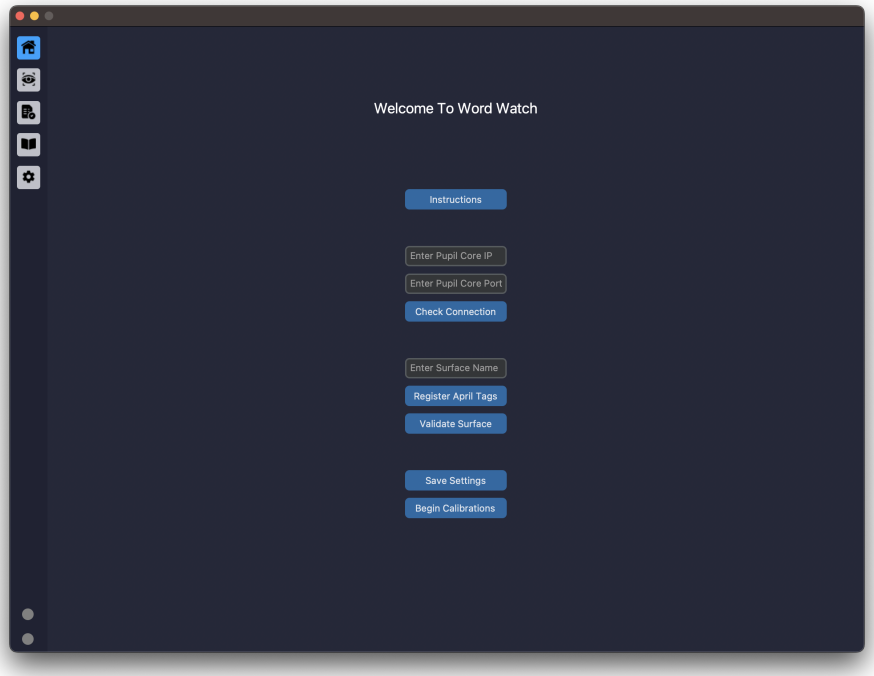

Figure 4: The Home Page

The Home Page is the first page that a user sees when Word Watch is launched. It contains all necessary fields and buttons for connecting the Pupil Core and registering Word Watch's April Tags in Pupil Capture's Surface Tracking plugin. The components of the Home page from top to bottom are as follows:

- **Instructions Button:** This button takes users to a document hosted that contains instructions on how to set up and use all of Word Watch's features.

- **IP and Port Fields:** In these fields, the user is required to enter the IP address and Port that are in their Network API plugin panel within Pupil Capture.
- **Check Connection to Pupil Button:** This button launches a background process that attempts to establish a connection with the Pupil Core using the entered IP address and Port. This process then updates the top status indicator of the navigation bar accordingly.
- **Surface Name Field:** Here, the user must enter the name of the surface registered on the Surface Tracker plugin. These names must be identical.
- **Register Surface April Tags Button:** This button displays the four April tags used by Word Watch so that a user can assign them to a surface within the Surface Tracker Plugin.
- **Check Surface Tracking Button:** This button also displays four April tags. It then launches a background process that checks if the correct surface data is being broadcast by the Network API. The bottom-most status indicator on the navigation bar is then updated.
- **Save Configuration Button:** This button saves the IP address, port number, and surface name, to a configuration file.
- **Begin Calibrations Button:** This button takes the user to the Calibration Page, where they can sequentially take the three calibrations. This button is only enabled if Word Watch is successfully connected to the Pupil Core and Word Watch's April Tags are correctly registered in Pupil Capture's Surface Tracking plugin.

4.1.3. Calibrations Page

As seen in Figure 5, the Calibrations Page provides users with access to all three calibrations required. The page displays the names of the three calibrations: Cross, Single-Word, and Saccade. Next to each calibration name, there is a button that takes the user to that specific calibration. The user must take the calibrations in order and retake them every time Word Watch is launched. This page also displays a completion status for each calibration. If the calibration has not been completed, the status will be marked with an "X" symbol. If the calibration has been completed, the status will be marked with a check symbol. Additionally, the page has a results section for each calibration. The results section for the Cross Calibration shows the user's accuracy in pixels. Here, the accuracy

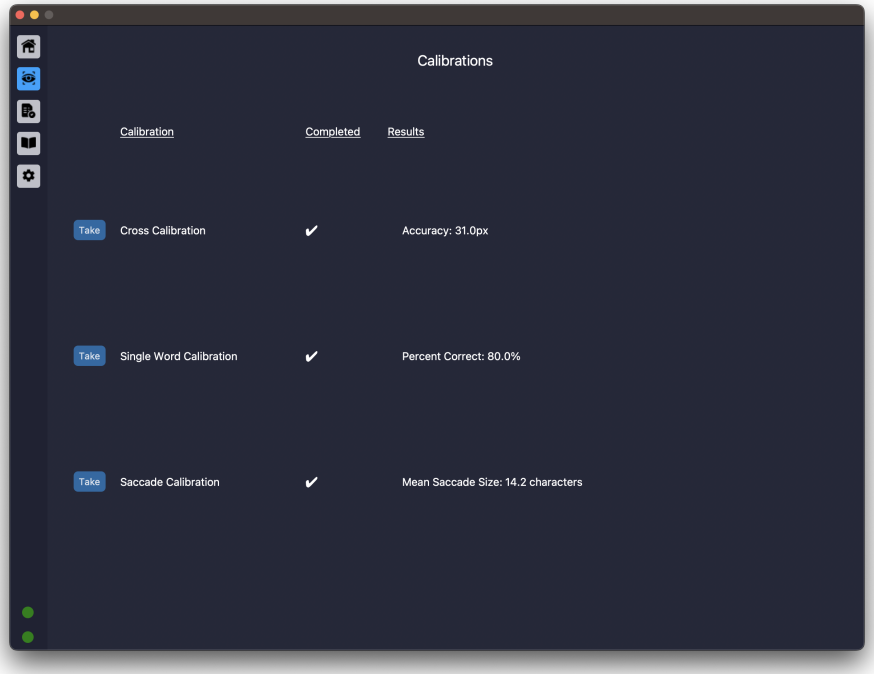

Figure 5: The Calibrations Page

is the average distance that fixation points were to the crosses in the calibration. The Single-Word Calibration results section displays the accuracy of the read-word predictions from the calibration. This accuracy is a percentage of words correctly identified as being looked at. Finally, the results section for the Saccade Calibration shows the user's average horizontal saccade size in characters. Additionally, users can retake these calibrations at any point, but must retake them sequentially. For example, if the user decides to retake the Cross Calibration, they must also retake the Single-Word and Saccade Calibrations. The following three sections will go into more detail about each of the three calibrations.

4.1.4. Cross Calibration

The goal of the Cross Calibration is to measure the accuracy surface coordinates of the fixation data that Word Watch receives. Once the user presses the green start button on the bottom right of the page, the calibration begins. The calibration consists of five crosses randomly displayed on a white canvas. Each cross is displayed one at a time. At first, crosses are displayed in black for one second to give the user time to fixate on the center of the cross. The crosses then turn green for two seconds and then disappear. There is a half a second delay between each cross. This calibration only uses

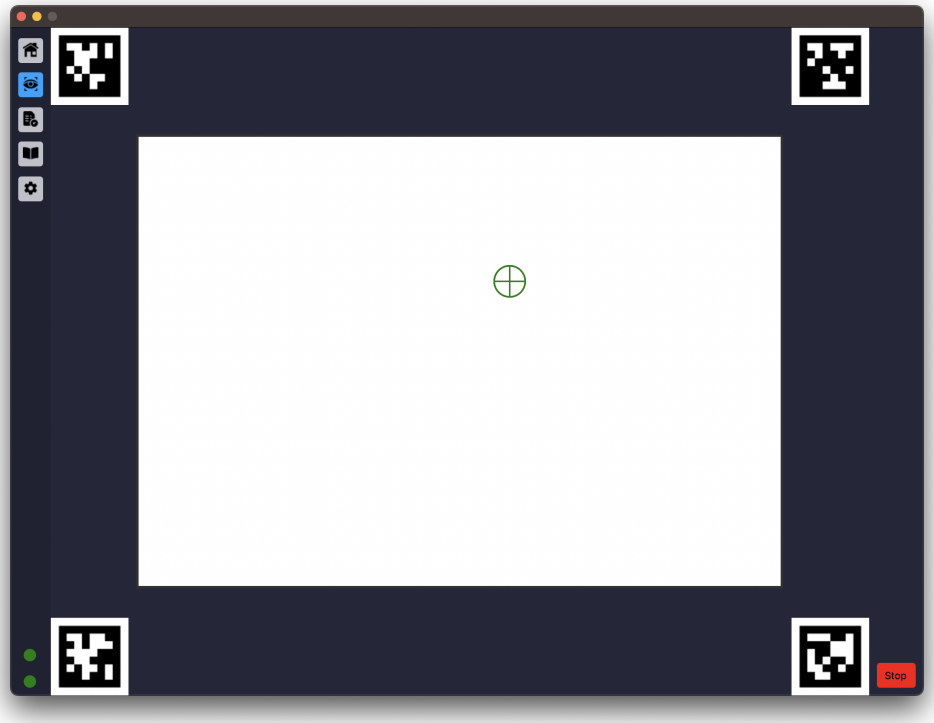

Figure 6: The Cross Calibration Page during a Calibration

gaze data received while each cross is green. After each cross has been displayed, Word Watch then computes the accuracy. The accuracy here is the average distance in pixels between every fixation data point and the center of the cross that the user was fixating on at that time. The text settings are then updated based on the calculate accuracy. The line spacing is set to the accuracy (in pixels) and the word spacing is set to half of the line spacing. After these calculations are complete, the user is then taken back to the calibration page, where the results are displayed.

4.1.5. Single-Word Calibration

The Single-Word Calibration aims to determine if the text settings established by the Cross Calibration are sufficient to handle words of varying lengths. The calibration page displays a canvas filled with text. The text is displayed using the text settings established by the Cross Calibration. Once the user presses the green start button on the bottom right of the page, the calibration begins. The calibration consists of five randomly selected words of varying lengths from the displayed text. Each word is displayed one at a time, with all other text hidden. Similar to the Cross Calibration, the word is displayed in black for one second to give the user time to fixate on the word. The

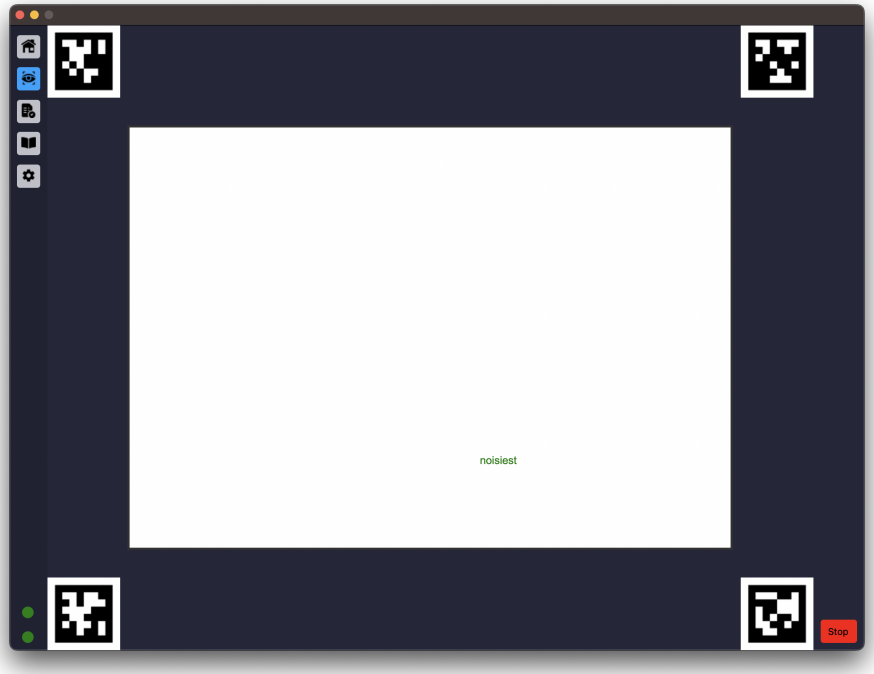

Figure 7: The Single-Word Calibration Page during calibration.

word then turns green for two seconds, with a half-second delay between each word. During the calibration, Word Watch records the fixation data received while each word is green. After each word has been displayed, this calibration then groups the fixation data by word and calculates the center of each group. We then find the word closest to this center and use this as our predicted word. If the predictions are less than 80% accurate, the line and word spacing are increased by 30%. After the calibration is finished, the user is returned to the Calibrations Page and the results section displays the percent correct. Users can retake this calibration multiple times to obtain more reliable text settings.

4.1.6. Saccade Calibration

Like the Single-Word Calibration, the Saccade calibration page displays a canvas filled with text. Once the user presses the green start button on the bottom right of the page, the calibration begins. Two random sentences are then selected from the displayed text and highlighted in red. The user reads the selected sentences and then presses the stop monitoring button. Next, the average horizontal distance (avhd) between subsequent fixations on the highlighted words is measured in characters. Finally, five predictions are made using the following thresholds: avhd, avhd$\pm$1, and

Figure 8: Saccade Calibration Page

avhd±2. The threshold that results in the predictions with the best F1 score is stored for use in the Test Read-Word Predictions Page and the Reading Monitoring Page.

4.1.7. Test Read-Word Predictions

The Test Read-Word Predictions Page is visually very similar to the Saccade Calibration page but uses a different page of text. The page highlights two sentences and makes predictions without changing the threshold set in Saccade Calibration. After predictions are made, the page changes the colors of words in order to help the user visualize the results (as seen in Figure 9). Finally, the True positive, False Positive, and False negative rates are stored in a JSON file labelled with the timestamp of the time of the test.

4.1.8. Reading Monitoring

The Reading Monitoring Page is where users can collect read-word data on their document of choice. This page uses the text settings and saccade threshold set by the three calibrations. As seen in Figure 10, the page is similar in appearance to the Test Read-Word Predictions page, with the addition of page numbers and previous and next buttons, allowing users to navigate long documents. Unlike the Test Read-Word Predictions page, however, sentences are not highlighted once monitoring begins. Read-word prediction is the same as in Test Read-Word Predictions, except

Figure 9: Visualization of Prediction Results In Test Read-Word Predictions Page. Dark green and dark blue respectively represent true positives and false positives found without the MWSF Algorithm. Light green and purple respectively represent true positives and false positive predictions made with the MWSF Algorithm. Red words are false negatives

that it is performed on a per-page basis. While a user is reading, the page stores the timestamps of every next and previous page button click. This information is then used to group the fixation data by page and make predictions per page. Once the user hits the stop button, the predictions are saved to JSON file with the current time and text (i.e what file is piece of text is being read) as the file-name. The file format can be seen in section 3.5. Like the Test Read-Word Predictions page, the Reading Monitoring page also allows users to visualize read-word data after monitoring has stopped. The Display Results button shows all read-word predictions made (if any) on every page using the color key described Figure 9.

4.1.9. Text Settings

The Text Settings are displayed in a separate popup window - triggered through the corresponding button on the navigation bar - that allows users to manually change how text is displayed. The window contains fields for font, font size, line spacing, and word spacing. All text settings are applied globally, meaning they affect all calibrations, the Test Read-Word Predictions page, and the Reading Monitoring page. It is recommended that users adjust font settings before calibration.

Figure 10: Reading Monitoring Page

Users are warned if they attempt to alter text settings after calibrations are complete, as doing so may decrease the accuracy of read-word predictions.

4.2. Multiprocessing and Communication With the Core

Multiprocessing is needed in this application in order to not interfere with the Tkinter's displaying of the GUI. When a calibration, test, or data collection session is started, a child process is spawned. This process then establishes a connection with the Pupil Core, filters for fixation data received on the surface, and stores that data in a multiprocessing queue. When the child process receives a stop signal from the parent process, the data in the multiprocessing queue is dumped. Status checking is performed similarly. The parent process periodically checks if the multiprocessing queue of the child process has any values. The child process repeatedly attempts to communicate with the Pupil Core (checking IP and port) and to receive surface information (checking if the surface is correctly registered). It then places a boolean value in the multiprocessing queue, which is read by the parent process and used to set the status.

5. Evaluation

In this section, we will evaluate the performance of several components of Word Watch. All evaluations were done by a single user, this paper's author. During evaluation, Word Watch was run exclusively on a 2022 Macbook Air with an Apple M2 CPU, 8GB of ram, macOS Ventura, and an external 27-inch monitor. *A Tale of Two Cities* by Charles Dickens was the text used for all tests.

5.1. Accuracy Loss via Surface Tracking

Pupil Lab's Surface Tracking plugin is essential to Word Watch's ability to determine where within the application a user is looking. However, the step of mapping fixation data from the Pupil Core's world camera onto a 2d surface can add inaccuracies. In this subsection, we evaluate the accuracy loss within Word Watch introduced by the surface tracking plugin. To do this, we compared the results of Pupil Capture's calibration with the results of Word Watch's Cross calibration. We successively performed the Pupil Capture calibration and the Cross Calibration five times. Both calibrations were done at a distance of 25 inches from a 27-inch monitor. The accuracy in this case is a measurement of the average difference in degrees of visual angle between a point on the monitor and the fixation location recorded from the Pupil Core. The Pupil Capture Calibration had a mean accuracy of 1.866°, while Word Watch's Cross Calibration had a mean accuracy of 2.191°. These results indicate that the surface tracking plugin introduces some inaccuracies into the system, but that the differences are relatively small.

5.2. Read-Word Identification Accuracy

It is important that Word Watch's read-word identification is as accurate as possible. Poor accuracy greatly hinders Word Watch's utility as a tool for data collection. In this section, we evaluate the accuracy of Word Watch's read-word identification using the Test Read-Word Prediction Test (see 4.1.7). It is important to note that we are not evaluating the accuracy of the subvocalization timing estimates in this subsection. Here we are evaluating whether words are correctly classified as read and as non-read. We completed 65 tests without using Word Watch's Cross and Single-Word

Calibration to modify text spacing. We then did an additional 65 tests after completing the Cross and Single-Word Calibrations which modified the text spacing. The text settings of these two groups can be seen below:

Text Setting	Without Calibration	With Calibration
Font	Arial	Arial
Font Size	15	15
Line Spacing	15 px	60 px
Word Spacing	5 px	25 px

Table 1: Text Settings for Read-Word Classification Tests

Each set of 65 tests contains a variety of reading speeds, ranging from around 100 words per minute to 350 words per minute. As expected, these adjustments to the text settings significantly increased read-word prediction accuracy. Because we do not care about the classification of true negatives, we chose to evaluate our predictions using an F1-score. The F1 score is calculated using the following equation:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{1}$$

As illustrated by Table 2, the F1 score of the tests that utilized calibration-set text settings was, on average, significantly higher than the tests which did not. Importantly, predictions made after calibrations were considerably more reliable, falsely marking a word as read only 2% of the time. The low false positive rates seen in both groups was expected. As mentioned previously, the Saccade Accomodation step (see 3.6.2) is quite conservative in its predictions of read words. The discrepancy between the two groups is likely due to improvements in the first step of prediction that solely uses fixation data. The increased spacing in the text after calibration results in higher quality read-word candidates produced by the first step. These candidates, in turn, improve the final predictions made by the Saccade Accomdation Algorithm (see 1). The improvements in reliability are also illustrated by the decrease in the STD of F1 scores in the tests after calibration.

Metric	Without Calibration	With Calibration
True Positives Proportion	0.32	0.67
False Positives Proportion	0.13	0.02
False Negatives Proportion	0.56	0.31
Mean F1 Scores	0.42	0.80
SD of F1 Scores	0.26	0.19

Table 2: A comparison of the relative proportions of True Positive, False Positive, and False negative predictions in the Without Calibration and With Calibration test runs and the Mean and STD of F1 Scores for each group

5.2.1. Reading Speed and Accuracy

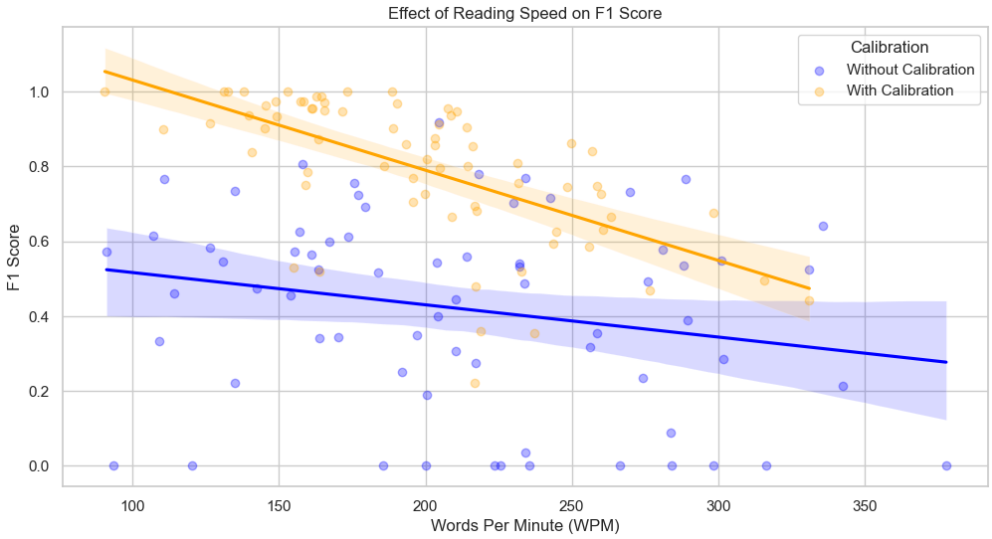

Figure 11: This regression plot shows the relationship between Reading Speed and F1 scores with and without calibration.

We also investigated the effect that reading speed had on F1 score across the Calibration and non-Calibration test groups. Unsurprisingly, as depicted in Figure 11, F1 score tends to decrease as reading speed increases. The F1 scores of the tests conducted after calibration exhibit a more distinct correlation with reading speed. This, again, can be attributed to the increased reliability resulting from the greater text spacing. Crucially, these result indicate that read-words can be accurately identified at reasonable reading speeds. It is somewhat obvious that increasing text spacing will also increase the accuracy of classification. After all, if there were only a Single-Word per page, it would be quite easy to identify which word a person was reading. This would, however,

greatly diminish the speed at which data could be collected. These results show that after Word Watch's calibrations have set text spacing and recorded horizontal saccade sizes, reading speeds of roughly 150 words-per-minute tend to result in read-word classifications with an F1 score of about 0.9. Although this speed still slower than normal reading speeds, which are usually between 175 and 320 words per minute [30], it nevertheless demonstrates that Word Watch can achieve a high accuracy in identifying read-words at a reasonable reading pace.

5.3. Subvocalization Timing Estimation Accuracy

In order to collect data for an SVR system, one must not only know *that* a word was read, but also know *when* that word was subvocalized. Validating the accuracy of our estimates for subvocalization timings was, unfortunately, outside of the scope of this project. Without having access to ground-truth information about the timing of subvocalizations, it is extremely challenging to evaluate the accuracy of predictions. Although our method of estimating subvocalization is rooted in prior research, we cannot evaluate this component of our produced read-word data without further testing. We discuss potential ways of assessing our subvocalization estimations in the Future Work section of this paper.

5.4. Reading Monitoring Page Performance

In this subsection, we will evaluate the performance of the reading monitoring page. More specifically, we will examine the maximum length of text that the page can accommodate and the maximum length of a recording session. Both of these metrics are limited by the memory of the device running Word Watch. In order to test evaluate the performance of the Reading Monitoring Page on large text files, we created a python script that generates text files of arbitrary lengths. This script takes the number of words as an input and generates a text file which contains that many words. Each word randomly created to be between 1 and 10 characters long, and each character is a random lowercase letter. In our testing, Word Watch was able to display text files which were 1 million words long before performance was impacted. Because both maximum text length and maximum session length are constrained by memory, longer text sizes will decrease and the maximum recording session

length and vice versa. In this evaluation, we assume that readers will not be reading thousands of pages per session. Consequently, we tested maximum session lengths using a text file of 5 thousand words, generated by the Python script described above. This is roughly 200 pages with typical text settings set by calibrations (see Table 1). We found that sessions could be a maximum length of 43 minutes before application performance begins to deteriorate. At a reading speed of 150 wpm, this equates to an idealized maximum of 6,300 read-word data points per session.

6. Limitations

There are several limitations of Word Watch that must be acknowledged. In this section, we will discuss these limitations in two sections: Implementation Limitations and Approach Limitations.

6.1. Implementation Limitations

The following limitations are not products of Word Watch's approach to data collection at a high-level, but are, instead, caused by how those approaches have been developed.

6.1.1. Reliance on External Software and Hardware

Because Word Watch heavily utilizes Pupil Labs' Capture software and Surface tracking and Network API plugins, future changes to these items could lead to unforeseen problems in the functioning of Word Watch. The reliance on the surface tracking plugin also adds several constraints on the positioning of the Word Watch window. During calibrations and reading monitoring, all four April Tags on the Word Watch window must be visible at all times. This means that the Word Watch window must remain as the front most application or surface tracking will fail, resulting in poor data quality. Similarly, any movement of the Word Watch application while calibration or data collection is ongoing can lead to inconsistencies in the surface coordinates of gaze and fixation data. This can also result in decreased data quality. Finally, Word Watch is completely reliant on the Pupil Core device. Problems will arise there are any defections with the device or if the device is unexpectedly disconnected from the computer running Word Watch.

6.1.2. Memory Requirements

Another limitation is that Word Watch does not log gaze and fixation data to a file as it receives

them from the network API. Instead, all gaze and fixation data for each session is stored in memory. We plan on addressing this limitation in a future version of Word Watch.

6.2. Approach Limitations

6.2.1. Sample Size :

All evaluations were done with a single participant, this paper's author. It may be the case that Word Watch's ability to identify read words does not generalize well to other individuals. To validate the read-word identification ability of Word Watch for a wider range of users, additional evaluations with multiple participants are necessary.

6.2.2. Read-Word Identification

As seen in the evaluation section, F1 accuracy drops considerably as reading speed increases, limiting the rate of data collection. The MWSF algorithm also produces a large amount of false negatives. This slows the rate at which data can be collected. Additionally, there are likely better heuristics or, alternatively, machine-learning approaches that could be used to classify read-words that would result in better performance.

6.2.3. Subvocalization Time Estimates

There are several limitations that are a result of our approach to estimate the timestamps of read words. As mentioned in the approach section, we make an assumption about the relationship between the lengths of fixations and the lengths of subvocalizations which may be incorrect and or imprecise. It may have been better to also consider the time it takes to say each word. A subvocalization of the word "queue", for example, is shorter than the length of the word would suggest. Additionally, our heuristic for determining the start of a subvocalization is, undoubtedly, an oversimplification. It is unclear if this approximation is accurate enough to allow for SVR models to be trained using our read-word data.

7. Future Work Section

7.1. Word Watch Future Development

There are several important steps that can be taken to further evaluate Word Watch and to improve the quality of the data it produces. As discussed in the limitations section, it is currently unclear if the quality of Word Watch's ground truth estimates for subvocalization timings are good enough to be used for SVR. To evaluate the viability of Word Watch's data, it would be useful to: (1) collect read-word data using Word Watch in conjunction with EMG data; (2) train a model using this data; and (3) compare the performance of this model to an identical model trained using data collected via prior research methods discussed in the Problem Background section. There are, however, improvements that could be made to Word Watch before evaluating it by performing SVR. For instance, a future version of Word Watch that incorporates both gaze and fixation data may be better able to deal with normal reading speeds.

7.2. Future Methods of SVR Data Collection

Another potential method for collecting ground truth data for SVR might be to monitor subvocalizations that occur during typing. This method would offer several advantages to Word Watch's approach. First, it would remove the need for an expensive eye-tracker, reducing the cost of SVR data collection. Second, there is far less ambiguity in the timing of typed words. Third, data collected using this method may be able to be more granular, corresponding to subvocalizations of parts of words, rather than words as a whole. Anecdotally, some people seem to subvocalize parts of words while typing (especially longer words that are more challenging to spell). This sounding out, however, seems to not occur during normal discursive thought. In this respect, subvocalizations that occur during reading may be more analogous to those which occur during thinking. Most of these points are highly speculative and would need to be evaluated more rigorously. Nevertheless, typing as a method for SVR data collection is an interesting future direction for research and could be potentially combined with Word Watch's reading-based approach.

8. Ethics

As with any emergent technology, SVR has a variety of potential benefits and harms. In this section, we will discuss the ethical implications of increasingly advanced SVR systems. Granting future technologies access to our internal monologues will undoubtedly radically transform the world. It is essential for researchers and society as a whole to consider the potential consequences of this technology while it is still in its infancy.

8.1. Potential Benefits

SVR technology has the potential to bring about significant improvements in various areas of human life. It may allow patients with neurodegenerative diseases to regain their ability to communication, improve our understanding of the human mind, and, more generally, enhance human-device interaction. There are undoubtedly also many benefits which are hard to foresee. In this subsection, we will explore some of these potential benefits in more detail.

8.1.1. Restoring communication for patients with neurodegenerative diseases

Current solutions for patients suffering from speech-impairing neurodegenerative diseases, such as amyotrophic lateral sclerosis (ALS), typically involve invasive, expensive surgery. While these technologies can be transformative for individuals who would otherwise have no way of communicating, they are still considerably slower at conveying information than speech. Researchers at Stanford have recently claimed to have achieved state-of-the-art performance in generating text from brain signals. Using a small electrode array implanted on the central sulcus, they recorded the signals from this region while ALS patients spoke. They then used this data to train a recurrent neural network, and attempted to predict text when the patients then imagined speaking. The researchers achieved a word error rate of 9.1% on a 50-word vocabulary one of 23.8% on a 125,000-word vocabulary and were able to generate text at 62 words per minute. While this is impressive and represents a 3x increase over the previous state of the art, 62 wpm is still less than half the speed of normal speech (162 wpm) [31]. More advanced SVR may allow for text creation at speeds

comparable to normal speech, and may even achieve this without the need for invasive surgeries. This could dramatically increase the quality of life for many patients.

8.1.2. Applications for Therapy and Psychology

Another potential positive application of SVR is in therapy and psychology at large. If non-invasive, accurate, and reliable SVR systems become widely available, individuals and therapists could use these systems to learn about their own habits of mind. Therapists could, for instance, use the technology to identify topics that trigger unhealthy rumination or anxiety in patients. Many behavioral techniques, such as Cognitive Behavioral Therapy (CBT) [32], aim to help individuals pay attention to, identify, and modify their own destructive patterns of thought. A competent SVR system may make these kinds of techniques easier. More broadly, advanced SVR could allow researchers in psychology to gain new insights about how the human mind functions.

8.2. Potential Risks

Although the potential benefits of SVR technology are significant, the technology also presents serious ethical concerns. The idea that unwanted parties could have access to our own private thoughts is frightening. If used without consent, future SVR systems would pose a serious threat to Western notions of privacy. We must, as a society, determine whether non-consensual uses of SVR will ever be permitted. In some cases, it may be tempting to use SVR systems non-consensually. SVR systems could, for instance, greatly reduce the error rates of court cases as the state of someone's internal monologue would likely bear some relation to their innocence or guilt. This benefit, however, may not justify the invasion of privacy it requires. In authoritarian regimes, capable SVR seems especially dangerous. Such systems would undoubtedly be useful Orwellian tools for suppressing dissent, monitoring citizens, maintaining power. As the development of SVR technology continues to advance, it is crucial for researchers, policymakers, and society as a whole to engage in discussions surrounding the ethical implications of this technology. By carefully considering both the potential benefits and risks, we can develop guidelines and regulations that ensure the responsible use of SVR, while maximizing its potential to improve lives and transform.

9. Summary

This project study presents Word Watch, an application designed to tackle the difficul- ties inherent in data collection for Subvocal Recognition (SVR). Word Watch offers researchers a novel approach to gathering data for SVR tasks. The application aims to facilitate the study of subvocalizations by accurately identifying when a word has been read during a reading task. In our experiments, Word Watch demonstrates promising results, accurately identifying when a word has been read at speeds slightly below the average reading pace. Despite these encouraging fin dings, further research is needed to assess the precision of the subvocalization time estimates generated by Word Watch. If the accuracy of these estimates can be refined, a future version Word Watch may be able to directly conduct large-scale data collection for SVR. We think that Word Watch demonstrates that reading-monitoring is potentially a viable route for SVR data collection and hope that other researchers pursue it further.

References

[1] Stephen Bottos and Balakumar Balasingam. An Approach to Track Reading Progression Using Eye-Gaze Fixation Points. URL: http://arxiv.org/abs/1902.03322, arXiv:1902.03322, doi:10.48550/arXiv.1902.03322.

[2] C. Jorgensen and K. Binsted. Web Browser Control Using EMG Based Sub Vocal Speech Recognition. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 294c–294c. doi:10.1109/HICSS.2005.683.

[3] Pattie Maes. AlterEgo: A Personalized Wearable Silent Speech Interface. URL: https://www.media.mit.edu/publications/alterego-IUI/.

[4] Pattie Maes. Non-Invasive Silent Speech Recognition in Multiple Sclerosis with Dysphonia. URL: https://www.media.mit.edu/publications/non-invasive-silent-speech-recognition-in-multiple-sclerosis-with-dysphonia/.

[5] David Gaddy and Dan Klein. Digital Voicing of Silent Speech. Comment: EMNLP 2020. URL: http://arxiv.org/abs/2010.02960, arXiv:2010.02960, doi:10.48550/arXiv.2010.02960.

[6] David Gaddy and Dan Klein. An Improved Model for Voicing Silent Speech. Comment: ACL 2021. URL: http://arxiv.org/abs/2106.01933, arXiv:2106.01933, doi:10.48550/arXiv.2106.01933.

[7] Maria L. Slowiaczek and Charles Clifton. Subvocalization and reading for meaning. 19(5):573–582. URL: https://www.sciencedirect.com/science/article/pii/S0022537180906283, doi:10.1016/S0022-5371(80)90628-3.

[8] Definition of Saccade. URL: https://www.merriam-webster.com/dictionary/saccade.

[9] Jochen Laubrock and Reinhold Kliegl. The eye-voice span during reading aloud. 6:1432. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4585246/, arXiv:26441800, doi:10.3389/fpsyg.2015.01432.

[10] Pupil Core - Eye tracking platform technical specifications - Pupil Labs. URL: https://pupil-labs.com/products/core/tech-specs/.

[11] Easy to use, small, portable eye tracker - Tobii Pro Nano. URL: https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-nano.

[12] Core - Pupil Capture. URL: https://docs.pupil-labs.com/core/software/pupil-capture/.

[13] Core - Surface Tracking Plugin. URL: https://docs.pupil-labs.com/core/software/pupil-capture/#surface-tracking.

[14] Core - Network API. URL: https://docs.pupil-labs.com/developer/core/network-api/.

[15] Tkinter — Python interface to Tcl/Tk. URL: https://docs.python.org/3/library/tkinter.html.

[16] Tom Schimansky. TomSchimansky/CustomTkinter. URL: https://github.com/TomSchimansky/CustomTkinter.

[17] gnikit. Tkinter-tooltip. URL: https://github.com/gnikit/tkinter-tooltip.

[18] Pupil-labs/apriltags. URL: https://github.com/pupil-labs/apriltags.

[19] Zeromq for python. URL: https://zeromq.org/languages/python/.

[20] MessagePack for Python. URL: https://github.com/msgpack/msgpack-python.

[21] NumPy. URL: https://numpy.org/.

[22] Scikit-learn: Machine learning in Python — scikit-learn 1.2.2 documentation. URL: https://scikit-learn.org/stable/.

[23] Valentin Lab. Colour. URL: https://github.com/vaab/colour.

[24] Module fitz — PyMuPDF 1.22.0 documentation. URL: https://pymupdf.readthedocs.io/en/latest/module.html.

[25] Aprilrobotics apriltags files · AprilRobotics/apriltag-imgs. URL: https://github.com/AprilRobotics/apriltag-imgs.

[26] Françoise Vitu, George W McConkie, Paul Kerr, and J. Kevin O'Regan. Fixation location effects on fixation durations during reading: An inverted optimal viewing position effect. 41(25):3513–3533. URL: https://www.sciencedirect.com/science/article/pii/S0042698901001663, doi:10.1016/S0042-6989(01)00166-3.

[27] Unix time. URL: https://en.wikipedia.org/w/index.php?title=Unix_time&oldid=1149173188.

[28] Core - Fixation Information. URL: https://docs.pupil-labs.com/core/terminology/#fixations.

[29] Flaticon - Free Icons and Stickers. URL: https://www.flaticon.com/https%3A%2F%2Fwww.flaticon.com%2F.

[30] Marc Brysbaert. *How Many Words Do We Read per Minute? A Review and Meta-Analysis of Reading Rate*. doi:10.31234/osf.io/xynwg.

[31] Francis Willett, Erin Kunz, Chaofei Fan, Donald Avansino, Guy Wilson, Eun Young Choi, Foram Kamdar, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. URL: https://www.biorxiv.org/content/10.1101/2023.01.21.524489v1, doi:10.1101/2023.01.21.524489.

[32] Barbara Olasov Rothbaum, Elizabeth A. Meadows, Patricia Resick, and David W. Foy. Cognitive-behavioral therapy. In *Effective Treatments for PTSD: Practice Guidelines from the International Society for Traumatic Stress Studies*, pages 320–325. The Guilford Press.